ANTENNAS SIMULATIONS

احمد محمود محمد الدقماق

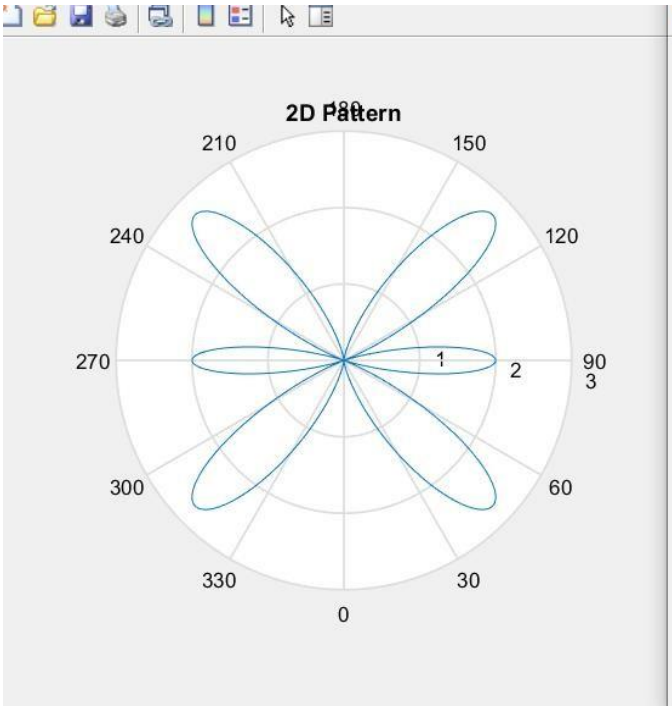
18010248

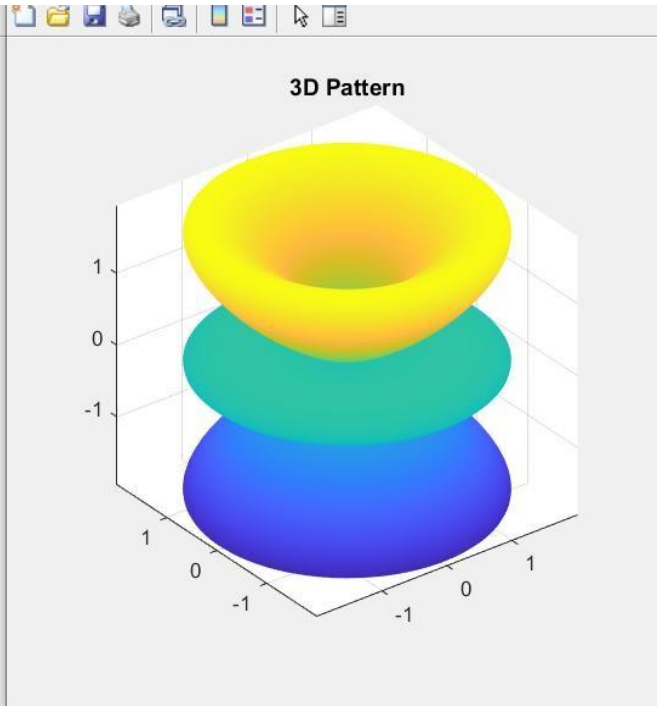
Prof.Dr. Said El-Khamy

PART1:LINEAR ANTENNA (DIPOLE OF GENERAL LENGTH

```
4    %% Part 1 : Linear antenna (dipole of general lenght)
5 -  Lambda_     = 1;
6 -  Betta       = (2*pi)/Lambda_;
7 -  Theta       = linspace(-pi,pi,360);
8 -  Phi         = linspace(-2*pi,2*pi,360);
9 -  L           = input('Enter The Lenght Of Dipole L, (L >1) : ');
10 - L           = L * Lambda_;
11 - En          = abs((cos( ( (Betta*L)/2 ).*cos(Theta)) - cos( ( (Betta*L)/2 ))./ sin(Theta));
12
13   % 2D Pattern
14 - figure(1);
15 - polar(Theta,En);
16 - view([90,90]);
17 - title('2D Pattern');
18   % 3D pattern
19 - phi_3D       = meshgrid(Phi);
20 - Theta_3D     = meshgrid(Theta);
21 - En_3D        = meshgrid(En);
22 - X           = En_3D .* sin(Theta_3D) .* cos(phi_3D');
23 - Y           = En_3D .* sin(Theta_3D) .* sin(phi_3D');
24 - Z           = En_3D .* cos(Theta_3D);
25
26 - figure(2);
27 - surf(X,Y,Z);
28 - shading interp;
29 - axis vis3d;
30 - axis equal;
31 - lighting gouraud;
32 - title('3D Pattern');
```

- L = 3

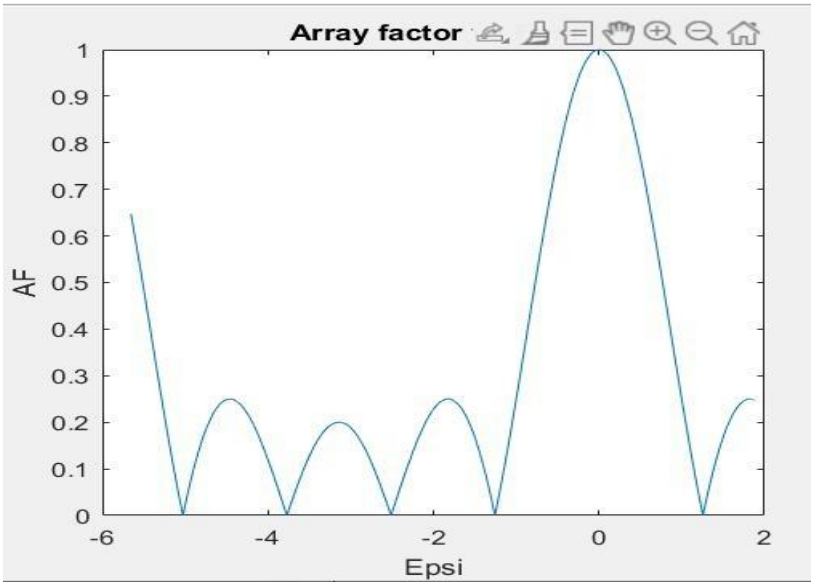


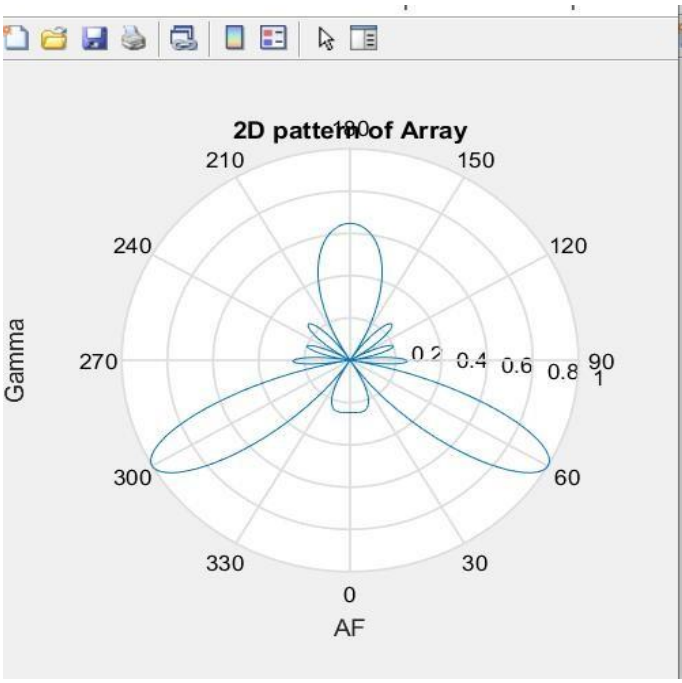


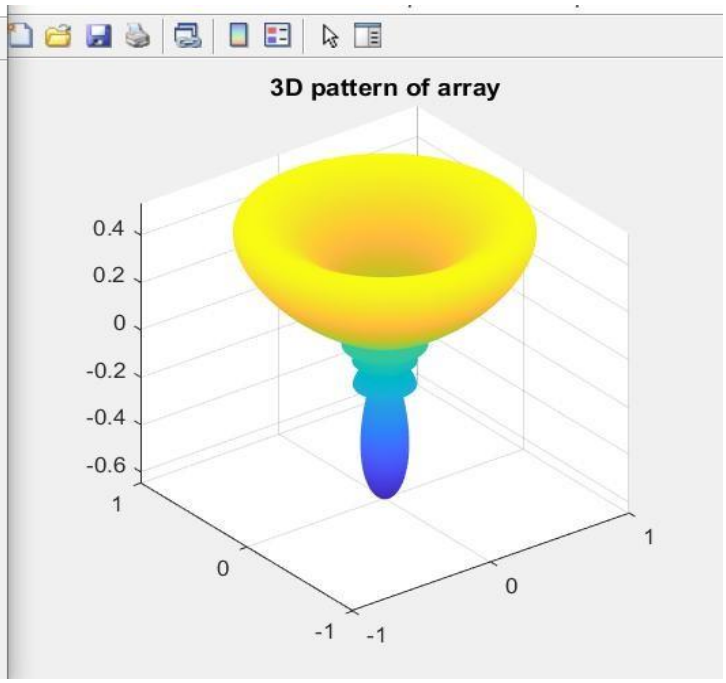
PART2:UNIFORM LINEAR ANTENNA ARRAY(ULA)

```
34      %% Part 2 : Uniform Linear Antenna Array(ULA)
35 -    Lambda_      = 1; Betta = (2*pi)/Lambda_;
36 -    d            = input('Enter the spacing (d >=0 ) : ');
37 -    d            = d * Lambda_ ;
38 -    N            = input('Enter the number of elements N (N>=0) : ');
39 -    Alpha        = input('Enter the progressive phase shift alpha : ');
40 -    Gamma        = linspace(-pi,pi,6000); Phi = linspace(-2*pi,2*pi,6000);
41 -    Epsi         = Betta*d*cos(Gamma) + Alpha;
42 -    AF           = abs( sin((N*Epsi)/2) ./ (N*sin(Epsi/2)) );
43
44      % 2D pattern
45 -    figure(1); plot(Epsi,AF);  title('Array factor vs Epsi');
46 -    xlabel('Epsi'); ylabel('AF');
47 -    figure(2); polar(Gamma,AF); view([90,90]); title('2D pattern of Array');
48 -    xlabel('Gamma'); ylabel('AF');
49      % 3D patter
50 -    Phi_3D     = meshgrid(Phi);
51 -    Gamma_3D   = meshgrid(Gamma);
52 -    AF_3D      = meshgrid(AF);
53 -    X          = AF_3D.*sin(Gamma_3D).*cos(Phi_3D');
54 -    Y          = AF_3D.*sin(Gamma_3D).*sin(Phi_3D');
55 -    Z          = AF_3D.*cos(Gamma_3D);
56 -    figure(3);
57 -    surf(X,Y,Z);
58 -    shading interp;
59 -    axis vis3d;
60 -    lighting gouraud;
61 -    title('3D pattern of array');
```

For L=3/5, N=5, α = -3Π/5



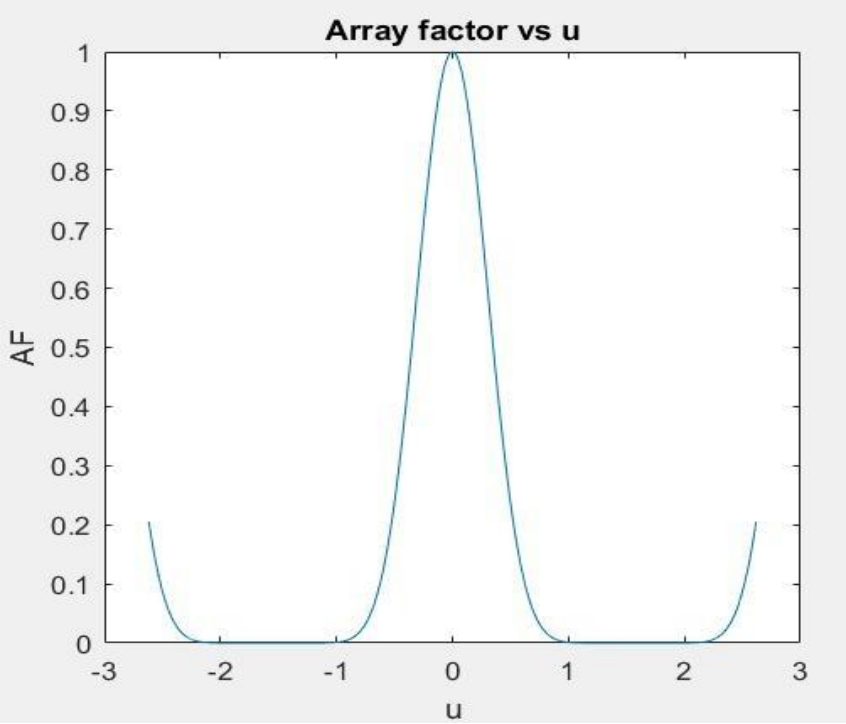

2D pattern of Array
210
150
240
120
270
90
300
60
330
30
0
180
0.2
0.4
0.6
0.8
1
Gamma
AF

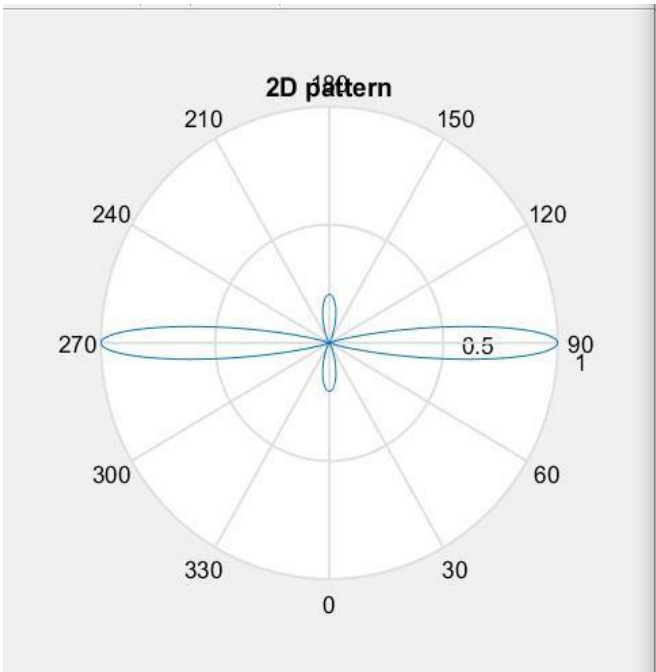

3D pattern of array
0.4
0.2
0
-0.2
-0.4
-0.6
1
0
-1
1
0
-1

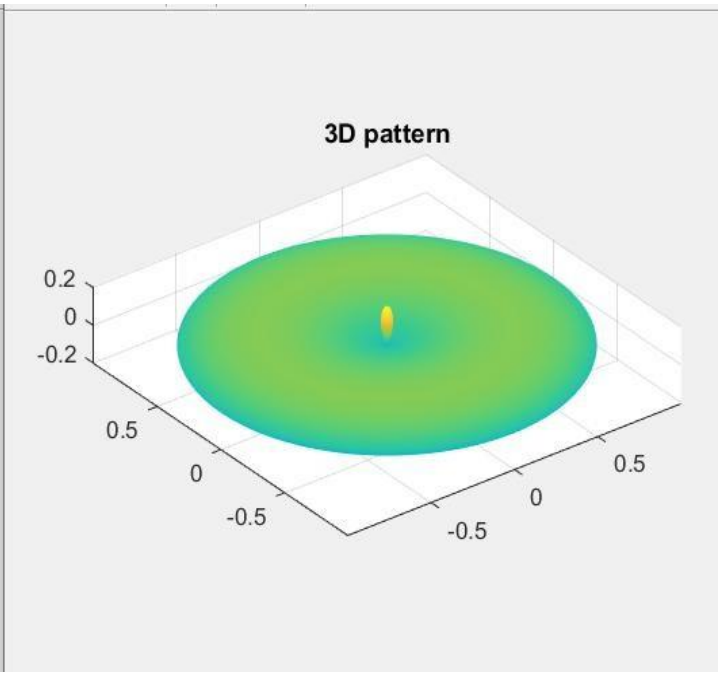
PART3:NONUNIFORMLY-FED LINEAR ANTENNAARRAY A.BINOMIAL ARRAYS

```
%% A-Binomial Arrays
Lambda_     = 1; Betta = (2*pi) / Lambda_;
d           = input('Enter the spacing d (d>=0) : ');
N           = input('Enter the number of elements N (N>=0) : ');
Alpha       = input('Enter the progressive phase shift alpha : ');
Theta       = linspace(-pi,pi,6000);
Phi         = linspace(-2*pi,2*pi,6000);
U           = (Betta*d*cos(Theta) + Alpha) / 2;
AF          = abs(cos(U).^(N-1));

% 2D pattern
figure(1); plot(U,AF); title('Array factor vs u');
xlabel('u'); ylabel('AF');
figure(2); polar(Theta, AF);
view([90 90]);
title('2D pattern');
% 3D pattern
Phi_3D   = meshgrid(Phi);
Theta_3D = meshgrid(Theta);
AF_3D    = meshgrid(AF);
X        = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
Y        = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
Z        = AF_3D.*cos(Theta_3D);

figure(3); surf(X,Y,Z);
shading interp;
axis vis3d;
axis equal;
lighting gouraud;
title('3D pattern');
```

For L=5/6, N=12, α = 0







PART3:NONUNIFORMLY-FED LINEAR ANTENNAARRAY B. DOLPH-TSHEBYSCEFF ARRAY

```
95      %% B-Dolph-Tschebysceff Arrays
96 -    Lambda_     = 1; Betta  = (2*pi) / Lambda_;
97 -    d           = input('Enter the spacing d (d>=0) : ');
98 -    N           = input('Enter the number of elements N (N>=0) : ');
99 -    Alpha       = input('Enter the progressive phase shift alpha : ');
100 -   M           = N - 1;
101 -   Ro          = input('Mainlobe to sidelobe level Ro (Ro>1) : ');
102 -   Zo          = cosh((1/M)*acosh(Ro));
103 -   Z           = linspace(-Zo,Zo,6000);
104 -   U_up        = acos(Z./Zo);
105 -   U_down      = -u_up;
106 -   U           = [U_down ; U_up];
107 -   Theta1      = acos(((2.*U_down)-Alpha)/(Betta*d));
108 -   Theta2      = -Theta1;
109 -   Phi         = linspace(-2*pi,2*pi,6000);
110 -   AF          = abs(cosh(M.*acosh(Z)));
111
112     % 2D Pattern
113 -   figure(1); plot(Z,AF); title('Array factor vs Z');
114 -   xlabel('Z'); ylabel('Array factor');
115 -   figure(2); polar(Theta1,AF);
116 -   hold on;
117 -   polar(Theta2, AF);
118 -   view([90 90]);
119 -   title('2D Pattern')
120     % 3D Pattern
121 -   Phi_3D   = meshgrid(Phi);
122 -   Theta_3D = meshgrid(Theta1);
123 -   AF_3D    = meshgrid(AF);
124 -   X        = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
125 -   Y        = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
126 -   Z        = AF_3D.*cos(Theta_3D);
127
128 -   figure(3); surf(X,Y,Z);
129 -   shading interp;
130 -   axis vis3d;
131 -   axis equal;
132 -   lighting gouraud;
133 -   title('3D Pattern')
```

For L=1, N=4, α = -Π, Ro=12